Binance Alpha 2.0 Liquidity Provision Program v.2025.10.01

Change Log

Date	Change
2025.09.25	Add endpoint “GET /sapi/v1/capital/alpha/config/getall” to Fetch alpha token & network info.
2025.08.27	To limit the accessibility of the Alpha market data to registered market makers, we will make "subscribed to listenKey stream" a precondition to subscribe to depth stream and bookTicker stream.The change is planned to be released on 2nd Sep.

Binance Alpha 2.0 integrates Binance Alpha directly within the Binance Exchange, bridging the gap between centralized (CEX) and decentralized (DEX) trading. It enables Binance.com users to access assets, liquidity, and volatility that are not directly available on Binance Spot or Future trading.

Here Binance Alpha 2.0 goes into new phases to allow users to enjoy more liquidity from market makers in addition to the current ones across decentralized exchanges on BSC, Ethereum, Solana, Base, and more.

Binance Alpha 2.0 Liquidity APIs

Besides the current User Interface for Instant and Limit orders, a new suite of APIs is available for market makers to send limit orders. All the limit orders will together join a call-auction-based trade and settlement system to trade. This system is defined with below features that make it different from the order book market Binance.com Spot trading provides:

1. Binance Alpha tokens can only be traded against stablecoins and major native tokens.
2. Only orders with limit prices are allowed, which are all good-till-cancel.
3. Orders are matched with periodic, automated call auctions. Different trading pairs may have different random auction time periods according to the blocking time of different blockchains that the tokens are on.

4. A price range is defined based on the live market prices of the tokens and a series of other parameters, such as the volatility of the token prices. The concluded auction price of each trading pair will be within this corresponding price range.

Similar to the current Alpha 2.0 users, allowed market makers can use their stablecoins and native tokens in their Binance.com 'Funding Wallet' to buy Alpha tokens; their bought tokens will be deposited into their 'Alpha Wallet'; they can sell these tokens into the stablecoins and native tokens back to their Funding Wallet.

Several groups of APIs are available to facilitate the full lifecycle of trades:

- Trade APIs to place, cancel, and query orders
- WebSocket APIs to get real-time feeds about order status in the system
- Fund APIs to withdraw Alpha tokens from Alpha Wallet, while market makers can also deposit Alpha tokens into specified addresses. (Binance.com listed tokens remain with the current deposit/withdrawal process and APIs.)

Please refer to the API documentation for both APIs.

MM quick start guide

1. Provide Binance master account UID and sub account UID to us, and we will help to whitelist the accounts(Kindly confirm with your POC that the KYC/KYB of the account is not associated with a restricted region.)
2. If you wish to restrict API access by IP, please provide a whitelist of up to 10 IP addresses (maximum due to security reasons). If no IP restriction is needed, you may leave this blank.
3. Deposit / Withdraw to master account, and use the sub account to trade (all alpha assets are in alpha wallet, and USDT/USDC are in funding wallet or alpha wallet of sub-accounts), can refer to FAQ funding flow of the place order.
4. All the trading symbols will be using alpha id, each alpha id corresponds to a specific token. The mapping can be found in https://www.binance.com/bapi/defi/v1/public/wallet-direct/buw/wallet/cex/alpha/all/token/list
5. All the available trading symbols can be found in the get exchange info API
   GET `/sapi/v1/alpha-trade/get-exchange-info`
6. Before place order, a user can subscribe to websocket to get user order push and market data push. User can generate a listen key via get listen key API
   POST `/sapi/v1/alpha-trade/get-listen-key`
   The listen key will expire in 1 hour, and a user can call the same API to extend the listen key for an extra hour. If the listen key is already expired, calling the API will return a new listen key
7. Connect websocket **wss://nbstream.binance.com/w3w/alpha,** and subscribe to one or multiple streams. Please note that only UI orders will be shown in the depth stream.

```
{"method":"SUBSCRIBE","params":["alpha_179usdt@depth",
"f79757ca63456c313d5fbd4106b5b09a"],"id":1}
where f79757ca63456c313d5fbd4106b5b09a is the listen key from the
get listen key API
```

8. Place order with place order API
   POST `/sapi/v1/alpha-trade/order/place`
   And observe websocket push
9. Please note that retail users probably place orders by referencing the index price shown in the UI, which is a weighted average price from DEX. This price is different from the execution price in this offchain orderbook. MM can get the index price through the listen key websocket stream, which is similar to UI's index price but is not exactly the same. This price is further aggregated and optimized, aiming to reflect a more accurate and stable market value.

Call Auction Match Logic

Examples

The chosen price level row would have ``*`` on the row where the price was concluded as the EP because of the column.

```java
// Example A: Choose the largest execution (Step 1)
-------------------------------------------------------------
SUM    SELL    PRICE    BUY    SUM    EXECUTION    IMBALANCE
300            100      150    150    150          -150
300            99              150    150          -150
300    250     98       150    300    300*         0
50     50      97              300    50           250
```

```java
// Example B: Choose the largest execution (Step 1)
-------------------------------------------------------------
SUM    SELL    PRICE    BUY    SUM    EXECUTION    IMBALANCE
300            100      150    150    150          -150
300            99       50     200    200          -100
300            98              200    200          -100
300    200     97       300    500    300*         200
100    100     96              500    100          400
```

```java
// Example C: the least abs surplus imbalance (Step 2)
-------------------------------------------------------------
SUM    SELL    PRICE    BUY    SUM    EXECUTION    IMBALANCE
1500           102      300    300    300          -1200
1500           101             300    300          -1200
1500           100      100    400    400          -1100
1500           99       200    600    600          -900
1500   250     98       300    900    900          -600
1250   250     97              900    900          -350
1000   1000    96              900    900          -100*
```

Java

```
// Example D: the least abs surplus imbalance (Step 2)
-------------------------------------------------------------
SUM    SELL    PRICE    BUY    SUM    EXECUTION    IMBALANCE
110            102      30     30     30           -80
110            101      10     40     40           -70
110            100             40     40           -70
110            99       50     90     90           -20
110    10      98              90     90           -20
100    50      97              90     90           -10*
50             96       15     105    50           55
50     50      95              105    50           55
```

Java

```
// Example E: if several prices have the same smallest absolute surplus amount
and are all negative, choose the closest to L to indicate sell pressure (Step
3)
-------------------------------------------------------------
SUM    SELL    PRICE    BUY    SUM    EXECUTION    IMBALANCE
50             102      10     10     10           -40
50             101             10     10           -40
50             100             10     10           -40
50             99              10     10           -40
50             98              10     10           -40
50             97       10     20     20           -30
50             96              20     20           -30
50     50      95              20     20           -30*
```

Java

```
// Example F: if several prices have the same smallest absolute surplus amount
and are all positive, choose the closest to H to indicate buy pressure (Step 3)
-------------------------------------------------------------
SUM    SELL    PRICE    BUY    SUM    EXECUTION    IMBALANCE
50             99       100    100    50           50*
50             98              100    50           50
50             97              100    50           50
50             96              100    50           50
50             95              100    50           50
```

```
50              94              100     50          50
50              93              100     50          50
50      50      92              100     50          50
```

Java

```
//Example G: The last price is 99. Choose the closest to the last trade price
99 (Step 4)
-------------------------------------------------------------------
SUM     SELL    PRICE     BUY     SUM     EXECUTION     IMBALANCE
50              100       25      25      25            -25
50              99                25      25            -25*
50      25      98                25      25            -25
25              97        25      50      25            25
25              96                50      25            25
25      25      95                50      25            25
```

Java

```
//Example H: The last price is 105. Choose the closest to the last trade price
100 (Step 4)
-------------------------------------------------------------------
SUM     SELL    PRICE     BUY     SUM     EXECUTION     IMBALANCE
50              100       25      25      25            -25*
50              99                25      25            -25
50      25      98                25      25            -25
25              97        25      50      25            25
25              96                50      25            25
25      25      95                50      25            25
```

```
```

Order Fill/Execution

After the execution price EP is concluded. Order match would happen in sequence of the price and time, i.e.

- Orders with best bid price would match with order with best ask price;
- If the orders on one price cannot be fully filled by the opposite orders for the orders with the same price, the orders would be selected based on the time they arrive into CAME.There can be partial filled orders on the leftover side.

Alpha APIs

Base url	https://api.binance.com/
API Key & Secret	Use the same ones from: https://www.binance.com/en/my/settings/api-management
Sign Requests	Refer to https://developers.binance.com/docs/margin_trading/general-info#signed-trade-user_data-and-margin-endpoint-security

Quote Assets (USER_DATA)

API Description

Fetch quote assets supported on the alpha market.

HTTP Request

GET `/sapi/v1/alpha-trade/get-from-asset`

Request Weight(IP)

1

Request Parameters

Name	Type	Mandatory	Description
recvWindow	Long	NO	
timestamp	Long	YES	

Response Example


```
["USDT","USDC"]
```


Fetch Token Info (USER_DATA)

API Description

Fetch alpha token & network info.

HTTP Request

GET `/sapi/v1/capital/alpha/config/getall`

Request Weight(IP)

1

Request Parameters

Name	Type	Mandatory	Description
recvWindow	Long	NO	
timestamp	Long	YES	

Response Example


```
[
    {
        "network": "SOL",
        "coin": "ALPHA_42",
        "name": "AVAAI",
        "symbol": "AVAAI",
```

```
        "entityTag": "alpha",
        "isDefault": true,
        "depositEnable": true,
        "withdrawEnable": true,
        "depositDesc": "",
        "withdrawDesc": "",
        "specialDepositTips": "",
        "specialWithdrawTips": "",
        "addressRegex": "^[1-9A-HJ-NP-Za-km-z]{32,44}$",
        "addressRule": "ADDRESS",
        "memoRegex": "",
        "withdrawFee": "28",
        "withdrawMin": "56",
        "withdrawMax": "9999999",
        "depositDust": "0.000001",
        "minConfirm": 1,
        "unLockConfirm": 0,
        "sameAddress": false,
        "estimatedArrivalTime": 0,
        "contractAddressUrl": "https://explorer.solana.com/address/"
    },
...
]
```


Exchange Info (USER_DATA)

API Description

Fetch all exchange info supported on alpha market.

HTTP Request

GET `/sapi/v1/alpha-trade/get-exchange-info`

Request Weight(IP)

1

Request Parameters

Name	Type	Mandatory	Description
recvWindow	Long	NO	
timestamp	Long	YES	

Response Example


```
{
    "timezone": "UTC",
    "assets": [
        {
            "asset": "USDT"
        },
        {
            "asset": "ALPHA_116"
        }
    ],
    "symbols": [
        {
            "symbol": "ALPHA_116USDT",
            "status": "TRADING",
            "baseAsset": "ALPHA_116",
            "quoteAsset": "USDT",
            "pricePrecision": 8,
            "quantityPrecision": 8,
            "baseAssetPrecision": 8,
            "quotePrecision": 8,
            "filters": [
                {
                    "filterType": "PRICE_FILTER",
                    "minPrice": "0.10000000",
                    "maxPrice": "100000",
                    "tickSize": "0.10000000"
                },
                {
                    "filterType": "LOT_SIZE",
```

```
            "stepSize": "0.00000100",
            "maxQty": "9000000",
            "minQty": "0.00000100"
        },
        {
            "filterType": "MAX_NUM_ORDERS",
            "limit": 200
        },
        {
            "filterType": "MIN_NOTIONAL",
            "minNotional": "5"
        },
        {
            "filterType": "MAX_NOTIONAL",
            "maxNotional": "100000"
        },
        {
            "filterType": "NOTIONAL",
            "minNotional": "5",
            "maxNotional": "100000"
        },
        {
            "filterType": "PERCENT_PRICE",
            "multiplierDown": "0.01",
            "multiplierUp": "100000"
        },
        {
            "filterType": "PERCENT_PRICE_BY_SIDE",
            "bidMultiplierUp": "100000",
            "askMultiplierUp": "100000",
            "bidMultiplierDown": "0.01",
            "askMultiplierDown": "0.01"
        }
    ],
    "orderTypes": [
        "LIMIT"
    ]
```

```
            }
        ]
}
```

Commission Fee (USER_DATA)

API Description

Fetch all commission fee info by symbol. Need to divide by 1 million .

HTTP Request

GET `/sapi/v1/alpha-trade/get-fee-rate`

Request Weight(IP)

1

Request Parameters

Name	Type	Mandatory	Description
recvWindow	Long	NO	
timestamp	Long	YES	
symbol	Long	YES	

Response Example

```
{
    "buyerCommission": 1000,
    "sellerCommission": 1000
}
```

Listen Key (USER_DATA)

API Description

Get listen key to subscribe websocket

HTTP Request

POST `/sapi/v1/alpha-trade/get-listen-key`

Request Weight(IP)

1

Request Parameters

Name	Type	Mandatory	Description
recvWindow	Long	NO	
timestamp	Long	YES	

Response Example


```
{
                              "listenKey":
        "df1f0c273c887e03f2d2ca12834994e8"
}
```

Place Order (TRADE)

API Description

Place alpha order

HTTP Request

POST `/sapi/v1/alpha-trade/order/place`

Request Weight(IP)

1

Request Parameters

Name	Type	Mandatory	Description
recvWindow	Long	NO	
timestamp	Long	YES	
baseAsset	String	YES	
quoteAsset	String	YES	
side	Enum	YES	BUY/SELL
quantity	Decimal	YES	
price	Decimal	YES	
clientOrderId	String	NO	^[a-zA-Z0-9-_]{32,36}$ It is recommended to use uuid
walletType	Enum	NO	Default FUNDING,if need to use alpha

			wallet,please input ALPHA

Response Example

```
{
    "orderId": "144193",
    "status": "S" // P:Processing, S:Success, F:Failure
}
```

Cancel Order (TRADE)

API Description

Cancel alpha order

HTTP Request

POST `/sapi/v1/alpha-trade/order/cancel`

Request Weight(IP)

1

Request Parameters

Name	Type	Mandatory	Description
recvWindow	Long	NO	
timestamp	Long	YES	
symbol	String	YES	

orderId	String	YES	

Response Example

```
{
    "orderId": "144193",
    "orderStatus": "CANCELED"
}
```

Cancel All Order (TRADE)

API Description

Cancel all alpha order

HTTP Request

POST `/sapi/v1/alpha-trade/order/cancel-all`

Request Weight(IP)

1

Request Parameters

Name	Type	Mandatory	Description
recvWindow	Long	NO	
timestamp	Long	YES	
symbol	String	NO	cancel all orders by symbol
baseAsset	String	NO	cancel all orders by baseAsset

if both symbol & baseAsset is not input, then we will cancel all the open orders

Response Example

```
{
    "success": true
}
```

Get Open Order (TRADE)

API Description

Get open orders

HTTP Request

GET `/sapi/v1/alpha-trade/order/get-open-order`

Request Weight(IP)

1

Request Parameters

Name	Type	Mandatory	Description
recvWindow	Long	NO	
timestamp	Long	YES	
symbol	String	NO	
side	Enum	NO	BUY/SELL

Response Example


```
[
    {
        "orderId": "144195",
        "symbol": "ALPHA_172USDT",
        "status": "NEW",
        "clientOrderId": "default_b3e7de38d9df4fc49bc7ba207f25",
        "price": "0.00000049",
        "avgPrice": "0",
        "origQty": "12245100.47",
        "executedQty": "0",

        "cumQuote": "0",
        "timeInForce": "GTC",
        "type": "LIMIT",
        "side": "BUY",
        "stopPrice": "0",
        "origType": "LIMIT",
        "time": 1743506038544,
        "updateTime": 1743506038544,
        "orderListId": "-1",
        "pageId": "18546215",
        "baseAsset": "ALPHA_172",
        "quoteAsset": "USDT"
    }
]
```


Get Order History (TRADE)

API Description

Get order history

HTTP Request

GET `/sapi/v1/alpha-trade/order/get-order-history`

Request Weight(IP)

1

Request Parameters

Name	Type	Mandatory	Description
recvWindow	Long	NO	
timestamp	Long	YES	
baseAsset	String	NO	
side	Enum	NO	BUY/SELL
orderStatus	String	NO	NEW,CANCELED,FILLED,PARTIALLY_FILLED any combination, separated by comma
startTime	Long	NO	
endTime	Long	NO	
limit	Long	NO	
pageId	Long	NO	pageId for pagination order:desc

Response Example


```
[
    {
        "orderId": "144195",
        "symbol": "ALPHA_172USDT",
        "status": "NEW",
        "clientOrderId": "default_b3e7de38d9df4fc49bc7ba207f25",
        "price": "0.00000049",
        "avgPrice": "0",
        "origQty": "12245100.47",
        "executedQty": "0",
```

```
            "cumQuote": "0",
            "timeInForce": "GTC",
            "type": "LIMIT",
            "side": "BUY",
            "stopPrice": "0",
            "origType": "LIMIT",
            "time": 1743506038544,
            "updateTime": 1743506038544,
            "orderListId": "-1",
            "pageId": "18546215",
            "baseAsset": "ALPHA_172",
            "quoteAsset": "USDT"
        }
    ]
```


Get Order Detail (TRADE)

API Description

Get order history

HTTP Request

GET `/sapi/v1/alpha-trade/order/get-order-detail`

Request Weight(IP)

1

Request Parameters

Name	Type	Mandatory	Description
recvWindow	Long	NO	

timestamp	Long	YES	
symbol	String	YES	
orderId	String	YES	

Response Example


```
[
    {
        "orderId": "144195",
        "symbol": "ALPHA_172USDT",
        "status": "NEW",
        "clientOrderId": "default_b3e7de38d9df4fc49bc7ba207f25",
        "price": "0.00000049",
        "avgPrice": "0",
        "origQty": "12245100.47",
        "executedQty": "0",
        "cumQuote": "0",
        "timeInForce": "GTC",
        "type": "LIMIT",
        "side": "BUY",
        "stopPrice": "0",
        "origType": "LIMIT",
        "time": 1743506038544,
        "updateTime": 1743506038544,
        "orderListId": "-1",
        "pageId": "18546215",
        "baseAsset": "ALPHA_172",
        "quoteAsset": "USDT"
    }
]
```

Get User Trades (TRADE)

API Description

Get user trades

HTTP Request

GET `/sapi/v1/alpha-trade/order/get-user-trades`

Request Weight(IP)

1

Request Parameters

Name	Type	Mandatory	Description
recvWindow	Long	NO	
timestamp	Long	YES	
baseAsset	String	NO	
side	Enum	NO	BUY/SELL
orderId	String	NO	
startTime	Long	NO	
endTime	Long	NO	
limit	Long	NO	
pageId	Long	NO	pageId for pagination order:desc

(startTime and endTime) required or orderId is required

Response Example

```
[
    {
        "symbol": "ALPHA_172USDT",
        "id": "708",
        "orderId": "96311",
        "tradeId": "708",
        "side": "BUY",
        "price": "0.00003132000",
        "qty": "122451.4700",
        "quoteQty": "3.83518004",
        "commission": "122.4515",
        "commissionAsset": "ALPHA_172",
        "time": 1743425641228,
        "pageId": "15942375",
        "buyer": true,
        "baseAsset": "ALPHA_172",
        "quoteAsset": "USDT",
        "orderType": "LIMIT",

        "lastTrade": false
    }
]
```


Kline Data (USER_DATA)

API Description

Fetch kline data supported on alpha market.

HTTP Request

GET `/sapi/v1/alpha-trade/market/klines`

Request Weight(IP)

1

Request Parameters

Name	Type	Mandatory	Description
symbol	String	YES	
interval	Enum	YES	interval: [1m, 3m, 5m, 15m, 30m, 1h, 2h, 4h, 6h, 8h, 12h, 1d, 3d, 1w, 1M]
startTime	Long	NO	
endTime	Long	NO	
limit	Integer	NO	
recvWindow	Long	NO	
timestamp	Long	YES	

Response Example

```
[
    1499040000000,
    "0.01634790",
    "0.80000000", // High
    "0.01575800",
    "0.01577100",
    "148976.11427815",
    1499644799999,
    "2434.19055334",

    308,

     "1756.87402397",
     "28.46694368",
     "0"
```

```
  ]
[
    [
        "1744977600000
        ",                  // Open time
        "0.19910000",       // Open
        "0.19910000",       // High
        "0.19910000",       // Low
        "0.19910000",       // Close
        "0.00000000",       // Volume
        "1744981199999
        ",                  // Close time
        "0.00000000",       // Quote asset volume
        "0",                // Number of trades
                            // Taker buy base asset volume, will remove
        "0.00000000",       the field,
legacy code
                            // Taker buy quote asset volume, will remove
        "0.00000000",       the field,
legacy code
        "0"                 // Ignore. will remove the field, legacy code
    ],
    [
        "1744981200000
        ",
       "0.19910000",
       "0.19910000",
       "0.19910000",
       "0.19910000",
       "0.00000000",
       "1744984799999",
       "0.00000000",
       "0",
       "0.00000000",
       "0.00000000",
       "0"
   ]
]
```

Ticker Info (USER_DATA)

API Description

Fetch 24hr ticker price change statistics

HTTP Request

GET `/sapi/v1/alpha-trade/market/ticker`

Request Weight(IP)

1

Request Parameters

Name	Type	Mandatory	Description
symbol	String	YES	
recvWindow	Long	NO	
timestamp	Long	YES	

Response Example


```
{
    "symbol": "ALPHA_175USDT",
    "priceChange": "0.05320000",
    "priceChangePercent": "36.463",
    "weightedAvgPrice": "0.16324830",
    "lastPrice": "0.19910000",
    "lastQty": "44.00000000",
    "openPrice": "0.14590000",
    "highPrice": "0.19910000",
```

```
    "lowPrice": "0.13170000",
    "volume": "1056.00000000",
    "quoteVolume": "172.39020000",
    "openTime": 1744105260000,
    "closeTime": 1744189233388,
    "firstId": 93,
    "lastId": 115,
    "count": 24
}
```


Symbol Price Ticker (USER_DATA)

API Description

Fetch symbol price ticker.

HTTP Request

GET `/sapi/v1/alpha-trade/market/ticker-price`

Request Weight(IP)

1

Request Parameters

Name	Type	Mandatory	Description
symbol	String	YES	
recvWindow	Long	NO	
timestamp	Long	YES	

Response Example

```
{
    "symbol": "ALPHA_175USDT",
    "price": "0.19910000",
    "time": 1744188294643
}
```


Aggregate Trade List (USER_DATA)

API Description

Fetch aggregate trade list

HTTP Request

GET `/sapi/v1/alpha-trade/market/agg-trades`

Request Weight(IP)

1

Request Parameters

Name	Type	Mandatory	Description
symbol	String	YES	
fromId	Long	NO	
startTime	Long	NO	
endTime	Long	NO	
limit	Integer	NO	
recvWindow	Long	NO	
timestamp	Long	YES	

Response Example

```
[
    {
        "a": 114, // aggTradeId
        "p": "0.19910000", // trade price, aka equilibrium price
        "q": "44.00000000", // quantity, trade total qty
        "f": 114, // firstBatchTradeId, orders are matches in batches at an
        random
interval, not immediately filled
        "l": 114, // lastBatchTradeId, orders are matches in batches, not
        immediately
filled
        "m": false // isBuyerMaker, legacy code not useful since
there is no maker/taker concept in batch trades, will be removed in
    },
    {
        "a": 115,
        "p": "0.19910000",
        "q": "44.00000000",
        "f": 115,
        "l": 115,
        "m": false
    }
]
```

Symbol Order Book Ticker (USER_DATA)

API Description

Fetch symbol order book ticker

HTTP Request

GET `/sapi/v1/alpha-trade/market/book-ticker`

Request Weight(IP)

1

Request Parameters

Name	Type	Mandatory	Description
symbol	String	YES	
recvWindow	Long	NO	
timestamp	Long	YES	

Response Example


```
{
    "symbol": "ALPHA_175USDT",
    "bidPrice": "0.00000000",
    "bidQty": "0.00000000",
    "askPrice": "0.13010000",
    "askQty": "47.00000000",
    "time": 1744984787471
}
```


Order Book Depth (USER_DATA)

API Description

Fetch order book depth. Please note that only UI orders will be shown.

HTTP Request

GET `/sapi/v1/alpha-trade/market/depth`

Request Weight(IP)

1

Request Parameters

Name	Type	Mandatory	Description
symbol	String	YES	
limit	Integer	NO	
recvWindow	Long	NO	
timestamp	Long	YES	

Response Example

```
{
    "lastUpdateId": 181404900,
    "symbol": "ALPHA_175USDT",
    "bids": [
        [
            "0.13010000", // PRICE
            "47.00000000" // QTY
        ]
    ],
"asks": [
        [
            "0.13010000", // PRICE
            "47.00000000" // QTY
        ]
    ],
    "t": 0, // Transaction time
    "e": 0 // Message output time
}
```

Alpha Asset Balance (USER_DATA)

API Description

Fetch alpha wallet asset balance

HTTP Request

GET `/sapi/v1/asset/get-alpha-asset`

Request Weight(IP)

1

Request Parameters

Name	Type	Mandatory	Description
recvWindow	Long	NO	
timestamp	Long	YES	

Response Example


```
[
    {
        "chainId": "56",
        "contractAddress": "0x9ec02756a559700d8d9e79ece56809f7bcc5dc27",
        "alphaId": "ALPHA_49",
        "cexAssetCode": "",
        "free": "1",
        "freeze": "0",
        "locked": "0",
        "withdrawing": "0",
        "amount": "1",
        "valuation": "0.00000003"
    },
    {
        "chainId": "",
        "contractAddress": "",
```

```
        "alphaId": "",
        "cexAssetCode": "USDC",
        "free": "2",
        "freeze": "0",
        "locked": "3",
        "withdrawing": "0",
        "amount": "5",
        "valuation": "4.9985"
    },
    {
        "chainId": "",
        "contractAddress": "",
        "alphaId": "",
        "cexAssetCode": "USDT",
        "free": "0.3",
        "freeze": "0",
        "locked": "0",
        "withdrawing": "0",
        "amount": "0.3",
        "valuation": "0.3"
    }
]
```


Alpha Token Mapping (USER_DATA)

API Description

Fetch alphaId & CA mapping info

HTTP Request

GET `/sapi/v1/alpha-trade/token/all/list`

Request Weight(IP)

1

Request Parameters

Name	Type	Mandatory	Description
recvWindow	Long	NO	
timestamp	Long	YES	

Response Example


```
[
    {
        "tokenId": "2CBEADBC555C72F33ACE06724CE40F09",
        "chainId": "146", // chain id
        "chainIconUrl":
"https://www.binance.com/image/admin_mgs_image_upload/20250313/2ee3cdc6-417a
ad6-a63d-975e2599cabe.png",
        "chainName": "Sonic",
        "contractAddress": "0x79bbf4508b1391af3a0f4b30bb5fc4aa9ab0e07c", //
        contract
address
        "name": "HeyAnon",
        "symbol": "Anon",
        "price": "5.404655700848514",
        "percentChange24h": "5.27",
        "volume24h": "867323.646352",
        "marketCap": "85137306.812400260985844151688",
        "fdv": "85137306.812400260985844151688",
        "liquidity": "579060.947307",
        "totalSupply": "15752586.570692",
        "circulatingSupply": "15752586.570692",
        "holders": "1",
        "decimals": 18,
        "listingCex": false,
        "hotTag": false,
        "cexCoinName": "",
        "canTransfer": false,
        "denomination": 1,
        "offline": true,
        "tradeDecimal": 8,
```

```
        "alphaId": "ALPHA_1000", //alpha token
        "offsell": false,
        "priceHigh24h": "5.491860723628665",
        "priceLow24h": "5.404655700848514",
        "onlineTge": false,
        "onlineAirdrop": false
    }
]
```


Alpha transfer between master and subaccount

Please refer to:
https://developers.binance.com/docs/sub_account/asset-management/Universal-Transfer
Set fromAccountType=alpha&toAccountType=alpha

Sample 01:
curl --location --request POST
'https://api.binance.com/sapi/v1/sub-account/universalTransfer?toEmail=toemail&fromAccountType=alpha&toAccountType=alpha&clientTranId=testtransferalpha&asset=USDC&amount=1&timestamp=*&signature=*' \
--header 'Content-Type: application/json' \
--header 'X-MBX-APIKEY: *

Sample 02:
curl --location --request POST
'https://api.binance.com/sapi/v1/sub-account/universalTransfer?toEmail=toemail&fromAccountType=spot&toAccountType=alpha&clientTranId=testtrans01&asset=USDC&amount=1&timestamp=1753418183854&signature=* \
--header 'Content-Type: application/json' \
--header 'X-MBX-APIKEY: *

Alpha Withdraw

API Description

Submit a withdrawal request.

HTTP Request

POST /sapi/v1/capital/alpha-withdraw/apply

Request Weight(IP)

1

Request param:

Name	Type	Required	Description
network	string	yes	Use “network” from GET /sapi/v1/capital/alpha/config/getall
alphaId	string	yes	alpha id <-> coin
contractAddress	string	yes	token contract address
address	string	yes	target address
addressTag	string	no	address memo
amount	decimal	yes	withdraw amount
clientOrderId	string	no	client id for withdraw
recvWindow	long	no	
timestamp	long	yes	

API Response:

```
{
    "id":"string" // withdraw id
}
```

Alpha Withdraw History

API Description

Fetch withdraw history.

HTTP Request

GET /sapi/v1/capital/alpha-withdraw/history

Request Weight(IP)

1

Request param:

Name	Type	Required	Description
alphaId	string	no	alpha id
clientOrderId	string	no	client id for withdraw
status	int	no	withdraw status (2:awaiting approval,3:rejected,4:processing,5:failed,6:success)
startTime	long	no	Default: 90 days from current timestamp
endTime	long	no	Default: present timestamp
offset	int	no	Default:0
limit	int	no	Default: 1000, Max: 1000
idList	string	no	id list returned in the response of POST /sapi/v1/capital/alpha-withdraw/apply, separated by ,
recvWindow	long	no	
timestamp	long	yes	

API Response:


```
[
    {
        "id": "string",
        "network": "string",
        "alphaId":"string",
        "contractAddress":"string",
        "coinName":"string",
        "address": "string",
        "addressTag": "string",
        "amount": 0,
        "txId": "string",
        "applyTime": "string",
        "completeTime": "string",
        "confirmNo": 0,
        "status": 0,
        "transactionFee": 0,
        "info": "string"
    }
```

]

Alpha Deposit History

API Description

Fetch deposit history.

HTTP Request

GET /sapi/v1/capital/alpha-deposit/history

Request Weight(IP)

1

Request Param:

Name	Type	Required	Description
alphaId	string	no	alpha id
txId	string	no	tx hash
status	int	no	depoist status, 0.pending, 1.success, 2.rejected, 5.awaiting approval, 6.credited but cannot withdraw
startTime	long	no	Default: 90 days from current timestamp
endTime	long	no	Default: present timestamp
includeSource	boolean	no	Default: false, return sourceAddressfield when set to true
offset	int	no	Default:0
limit	int	no	Default:1000, Max:1000
recvWindow	long	no	
timestamp	long	yes	

API Response:


```
[
    {
        "id": "string",
        "network":"string",
        "alpahId":"string",
        "contractAddress":"string",
        "coinName":"string",
        "address": "string",
        "addressTag": "string",
        "amount": 0,
        "txId": "string",
        "completeTime": 0,
        "confirmationNo": "string",
        "insertTime": 0,
        "sourceAddress": "string",
        "status": 0,
        "unlockConfirm": 0
    }
]
```


Alpha Fetch deposit address

API Description

Fetch deposit address.

HTTP Request

GET /sapi/v1/capital/alpha-deposit/address

Request Weight(IP)

1

Request Param:

name	type	required	description
network	string	yes	Use “network” from GET /sapi/v1/capital/alpha/config/getall
recvWindow	long	no	
timestamp	long	yes	

API Response: `


```
{
```

```
    "network": "BTC",
    "address": "1HPn8Rx2y6nNSfagQBKy27GB99Vbzg89wv",
    "tag": ""
}
```

Important: Only Alpha assets should be deposited to the address returned by this endpoint. Depositing any other assets may result in permanent loss.

Websocket

wss://nbstream.binance.com/w3w/alpha

Subscribe one/multiple stream(s)

```
{"method":"SUBSCRIBE","params":["alpha_111usdt@depth",
"f79757ca63456c313d5fbd4106b5b09a"],"id":1} //
f79757ca63456c313d5fbd4106b5b09a is the listen key
```

Received data will be in json format, the stream data will be inside the ‘data’ field.
`{"stream":"alpha_1111usdt@depth","data": <json_data>}`

Important:

Starting from September 2nd, before subscribing to any depth or bookTicker stream, the WebSocket connection must first be subscribed to a valid listenKey stream (MM Account listenKey) :

- ★ The listenKey must remain valid at the time of subscription.
- ★ You can include the listenKey in the same SUBSCRIBE request together with other streams, or you can send it in a separate SUBSCRIBE request first.
- ★ If no valid listenKey is subscribed, new subscription requests will be rejected with:
  `{"id":1,"error":{"code":2,"msg":"Invalid request: not authorized"}}`

Available streams ‘data’ format (all <symbol> in stream names are in lowercase)

<symbol>@miniTicker

```
{
  "e": "24hrMiniTicker",
  "E": 1676980794596, // timestamp
  "s": "ALPHA_111USDT",
  "c": "265.000000000", // lastPrice
  "o": "265.000000000", // openPrice
  "h": "265.000000000", // highPrice
  "l": "265.000000000", // lowPrice
  "v": "7.037735", // volume
  "q": "1864.999775000" // quoteVolume
}
```


<symbol>@ticker

```
{
  "e": "24hrTicker",
  "E": 1676982939357,
  "s": "ALPHA_111USDT",
  "p": "0.000000000",    // priceChange
  "P": "0.00",           // priceChangePercent
  "w": "265.00000000",   // weightedAvgPrice
  "c": "265.000000000",  // lastPrice
  "Q": "0.339615",       // lastQty
  "o": "265.000000000",  // openPrice
  "h": "265.000000000",  // highPrice
  "l": "265.000000000",  // lowPrice
  "v": "8.037735",       // volume
  "q": "2129.999775000", // quoteVolume
  "O": 1676899380000,    // openTime
  "C": 1676982939337,    // closeTime
  "F": 1381,             // firstTradeId
  "L": 1397,             // lastTradeId
  "n": 17                // numOfTrades
}
```

<symbol>@aggTrade

```
{
  "e": "aggTrade",
  "E": 1676985687671, // currentTimeMillis
  "a": 1106,          // aggTradeId
  "s": "ALPHA_111USDT",
  "p": "265.000000000", // price
  "q": "0.660385", // quantity
  "f": 1398,       // firstTradeId
  "l": 1398,       // lastTradeId
  "T": 1676985687498, // lastTransactTime
  "m": false      // isBuyerMaker
    (not useful, since alpha limit trading has no maker taker concept)
}
```

<symbol>@trade

```
{
  "e": "trade",
  "E": 1677039110296, // currentTimeMillis
  "T": 1677039110268, // timeStampOfExecutionReport
  "s": "ALPHA_111USDT",
  "t": 1406,          // batch trade id, all trades with same symbol
auction share same value
 (trades are executed in batch because of the auction matching
mechanism)
  "p": "265.500000000", // fillPrice
  "q": "0.100000",      // fillQty
  "m": true            // isBuyerMaker
(not useful, since alpha limit trading has no maker taker concept)
```

```
}
```

Depth Diff: <symbol>@depth

Please note that only UI orders will be shown.

```
{
    "e": "depthUpdate",
    "E": 1677038400454,  // currentTimeMillis
    "T": 1677038400160,  // transactTime
    "s": "ALPHA_111USDT",
    "U": 306910,         // firstUpdateId
    "u": 306910,         // lastUpdateId
    "pu": -1,            // prevUpdateId
    "b": [],             // bid
    "a": [               // ask
      [
        "300.000000",    // price
        "0.20"           // qty
      ]
    ]
  }
```

Depth Diff Custom Interval: <symbol>@depth@0ms available values: [0ms, 100ms, 500ms]

Please note that only UI orders will be shown.

```
{
    "e": "depthUpdate",
    "E": 1677038400454,  // currentTimeMillis
    "T": 1677038400160,  // transactTime
    "s": "ALPHA_111USDT",
    "U": 306910,         // firstUpdateId
    "u": 306910,         // lastUpdateId
    "pu": -1,            // prevUpdateId
    "b": [],             // bid
    "a": [               // ask
      [
        "300.000000",    // price
        "0.20"           // qty
      ]
    ]
  }
```

Depth Partial: <symbol>@depth5 available values: [5, 10, 20, 50]

Please note that only UI orders will be shown.


```
{
  "e": "depthUpdate",
  "E": 1677055009172, // currentTimeMillis
  "T": 1677055008894, // transactTime
  "s": "ALPHA_111USDT",
  "U": 30608390,      // firstUpdateId
  "u": 30608390,      // lastUpdateId
  "pu": 30608383,     // prevUpdateId
  "b": [
    [
      "265.500000000",
      "1.600000"
    ],
    [
      "265.000000000",
      "2.339615"
    ],
    [
      "256.000000000",
      "1.325781"
    ],
    [
      "255.500000000",
      "1.000000"
    ],
    [
      "255.000000000",
```

```
      "5.000000"
    ]
  ],
  "a": [
    [
      "266.000000000",
      "1.924814"
    ],
    [
      "275.000000000",
      "3.000000"
    ]
  ]
}
```

<symbol>@bookTicker

```
  {
    "u": 30608330,   // update id
    "e": "bookTicker",
    "s": "ALPHA_111USDT",
    "b": "265.500000000", // bestBidPrice
    "B": "2.000000",      // bestBidQty
    "a": "266.000000000", // bestAskPrice
    "A": "1.962407",      // bestAskQty
    "T": 1676989489943,   // lastTransactionTime
    "E": 1676989489958.   // currentTimeMillis
   }
```

<symbol>@bookTicker

```
{
  "u": 30608330,   // update id
  "e": "bookTicker",
  "s": "ALPHA_111USDT",
  "b": "265.500000000", // bestBidPrice
  "B": "2.000000",      // bestBidQty
  "a": "266.000000000", // bestAskPrice
  "A": "1.962407",      // bestAskQty
  "T": 1676989489943,   // lastTransactionTime
  "E": 1676989489958.   // currentTimeMillis
 }
```

!bookTicker (all symbols' book tickers update)

```
[{
  "u": 30608330,
  "e": "bookTicker",
  "s": "ALPHA_111USDT",
  "b": "265.500000000",
  "B": "2.000000",
  "a": "266.000000000",
  "A": "1.962407",
  "T": 1676989489943,
  "E": 1676989489958
}]
```

!miniTicker@arr (all symbols' mini tickers update)


```
[
    {
      "e": "24hrMiniTicker",
      "E": 1677038094077,
      "s": "ALPHA_111USDT",
      "c": "265.500000000",
      "o": "265.000000000",
      "h": "266.000000000",
      "l": "265.000000000",
      "v": "10.997978",
      "q": "2916.151763000"
    }
  ]
```

!ticker@arr (all symbols' tickers update)


```
[
  {
    "e": "24hrTicker",
    "E": 1677041026812,
    "s": "ALPHA_111USDT",
    "p": "0.500000000",
    "P": "0.189",
    "w": "265.16264795",
    "c": "265.500000000",
    "Q": "0.100000",
    "o": "265.000000000",
    "h": "266.000000000",
    "l": "265.000000000",
    "v": "11.297978",
    "q": "2995.801763000",
    "O": 1676980020000,
    "C": 1677041026788,
    "F": 1384,
    "L": 1408,
    "n": 25
  }
]
```

listen key stream

(listen key will expire in 1 hour, you can call the get listen key API to refresh the listen key. If the listen key not expired when calling the get listen key API, same listen key will be return and get extended 1 hour)

```
// auction window push
{
  "symbols": [
    "ALPHA_172USDC",
    "ALPHA_22USDC"
  ],
  "auctionWindowEndTimes": [
    1742984540012,
    1742984541783,
    1742984543178,
    1742984544736,
    1742984546300
  ],
  "batchId": 15917,
  "e": "auctionWindowUpdate"
}
// order push
{
  "s":"ALPHA_130USDT",   // symbol
  "c":"Xzh0gnxT41PStbwqOtXnjD",  // client order id
  "S":"SELL",   // order direction
  "o":"LIMIT",   // order type
  "f":"GTC",   // Time in force
  "q":"10.001000",   // Order quantity
  "p":"19.1000000000",   // Order price
  "ap":"19.0999999955550656",  //average price
```

```
  "P":"0",  //stop price
  "x":"TRADE",   // Current execution type
  "X":"PARTIALLY_FILLED",   // Current order status
  "i":"27",   // Order ID
  "l":"1",    // Last executed quantity
  "z":"8.999000",   // Cumulative filled quantity
  "L":"19.1000000000",   // Last executed price
  "n":"0.00382000",   // Commission amount, absent if no commission
  "N":"ALPHA_130",   // Commission asset, absent if no commission
  "T":1649926447190,,   //Trasanction Time
  "t":"18",   // batch trade id, all trades with same symbol auction
share same value
 (trades are executing in batch because of the auction matching
mechanism)
  "m":true,   // is this trade the maker side?
(not useful, since alpha limit trading has no maker taker concept)
  "ot":"LIMIT", //original order type
  "O":0,   // Order creation time
  "Z":"171.88089996",   // Cumulative quote asset transacted quantity
  "u":1649926447190,   //Trasanction Time, same as T
  "Y":"19.1000000000000000",   // Last quote asset transacted quantity
(i.e. lastPrice * lastQty)
  "Q":"0",   // Quote Order Qty
  "e":"executionReport",   // event
  "E":1649926447209, // event time,
  "ba": ALPHA_130, // base asset
  "qa": USDT // quote asset
  "id":2056857607 //pageId
}
// index price push
```

```
{
"indexPriceInfo": [
  {
    "tokenId": "ALPHA_131",
    "price": "0.00070728745835454 17",   // index price
    "upperBand": "0.000742651831272268",  // final matching price will be less than or equal to the upper band price
    "lowerBand": "0.000671923085436814",  // final matching price will be greater than or equal to the lower band price
    "time": 1745413782420
 },
 {
   "tokenId": "ALPHA_152",
   "price": "152.86007924947654",
   "upperBand": "160.503083211950367",
   "lowerBand": "145.217075287002713",
   "time": 1745413782420
 }
],
"e": "indexPriceUpdate"
}
// listen key expired event
{
"E": 1746199438136,
"e": "listenKeyExpired"
}
```


<symbol>@kline_<interval> interval available values: [1m, 3m, 5m, 15m, 30m, 1h, 2h, 4h, 6h, 8h, 12h, 1d, 3d, 1w, 1M]

```
{
  "e": "kline",
  "E": 1677132298231,   // currentTimeMillis
  "s": "ALPHA_111USDT",
  "k": {
    "t": 1677132000000, // openTime
    "T": 1677132899999, // closeTime
    "s": "ALPHA_111USDT",
    "i": "15m",         // interval
    "f": 28,            // firstBatchTradeId
    "L": 28,            // lastBatchTradeId
    "o": "260.000000000", // open
    "c": "260.000000000", // close
    "h": "260.000000000", // high
    "l": "260.000000000", // low
    "v": "0.100000",      // volume
    "n": 1,               // numberOfTrades
    "x": false,           // isClosed
    "q": "26.000000000",  // quoteVolume
    "V": "0.000000",      // takerBuyVolume
(not useful, since alpha limit trading has no maker taker concept)
    "Q": "0",             // takerBuyQuoteVolume
(not useful, since alpha limit trading has no maker taker concept)
    "B": "0"              // always 0
  }
}
```

How to manage a local order book correctly

1. Open a stream to **wss://nbstream.binance.com/w3w/alpha** Subscribe symbol depth ws by sending message **{"method":"SUBSCRIBE","params":["ALPHA_111USDT@depth"],"id":1}**

2. Buffer the events you receive from the stream. For same price, latest received update covers the previous one.
3. Get a depth snapshot from SAPI get depth
4. Drop any event where **u** is < **`lastUpdateId`** in the snapshot.
5. The first processed event should have **U** <= **`lastUpdateId`** **AND** **u** >= **`lastUpdateId`**.
6. While listening to the stream, each new event's **pu** should be equal to the previous event's **u**, otherwise initialize the process from step 3.
7. The data in each event is the **absolute** quantity for a price level.
8. If the quantity is 0, **remove** the price level.
9. Receiving an event that removes a price level that is not in your local order book can happen and is normal.

FAQ

Q) Only Binance Alpha tokens are going to be available?
A: Yes, and once Alpha tokens are listed in Spot, they will be delisted

Q) Will MMs be matched only with retail or other MMs?
A: Everyone can match everyone for now

Q) If two MMs have a limit order with the same price, how will execution logic work?
A: price first and then time

Q) From the doc it seems that external withdrawals of Alpha tokens are allowed, but will it be allowed to withdraw those to clients' Fireblocks address? to what addresses we can send?
A: it follows same whitelist address of binance exchange

Q) Will there be any trading limits?
A: exchange_info API will return most of the limits

Q) all data seems to be only accessible with api key, correct?
A: yes

Q) For the order books, I assume we will see crossed orderbook in the depth feeds, is there also a feed that publishes the indicative uncross price?

A: yes, it's periodically call auction base, you will see crossed orderbook. Please refer to the auction logic for potential price based on the orderbook.

Q) is order placement cancellation solely accessible through rest?
A: Yes, for now.

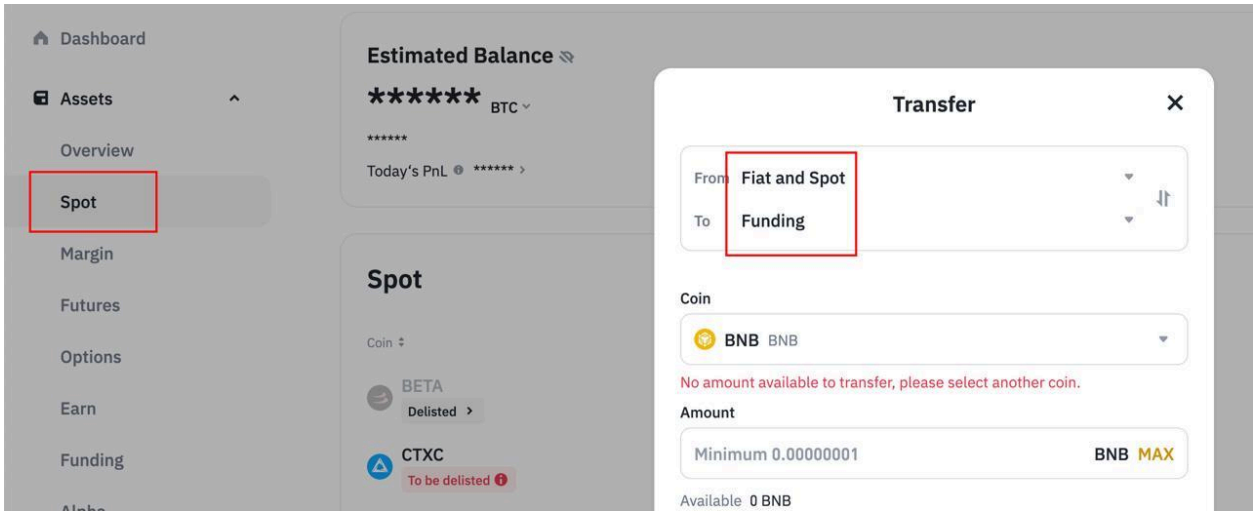
Q) what api keys are supported? also ed25519?
A: Whatever is supported by binance exchange

Q) What's the latency and rate limit?
A: Latency p99 67ms, rate limit: per ip max 12k/mins

Q) Does market data websocket push require authentication?
A: No, only "listen key" stream requires get listen key through API that requires signature

Q) How frequent is the market data stream update?
A: Most of the streams are pushed only when there is data change, like Depth Diff, Depth Partial, ticker etc. Kline stream will push when there is data change and also push every kline interval (1m, 5m, … etc).

Q) Where should I deposit USDT/USDC in order to trade?
A: Funding wallet, you can transfer from other wallet as below



Q) Where should I deposit the alpha asset to trade?
A: You can get the deposit address via the Alpha Fetch deposit address API and deposit alpha asset. After deposit, the alpha asset should be in the alpha wallet.

Q) What's the funding flow of placing an order?

A: 1.For a buy order, it will deduct the “free” USDT/USDC from the funding wallet or Alpha wallet (depending on which walletType is selected when placing the order).

2. After the deduction is successful, it freezes to the field “locked” in Alpha Wallet. Steps 1 and 2 are not atomic, and we query the balance information from the db, there may be some delay issues here, which will cause the freeze not to be added after the free deduction is completed. You can delay 100ms to query

1. For sell order, make sure you have the alpha asset in the alpha wallet. You can get the deposit address via the Alpha Fetch deposit address API
2. After trade, the USDT/USDC received from trade will be in the funding wallet or alpha wallet(depending on which walletType is selected when placing the order).

(Alpha token only exists in alpha wallet, USDC/USDT can only be used in funding wallet, but when you place the order, we will deduct USDC/USDT and freeze it in alpha wallet.
If you cancel the order, we will unfreeze the USDC/USDT in alpha wallet and return back to funding wallet)

Q) Is there trade level global unique id?
A: Yes, can refer to “pageId”

Q) Place order clientOrderId parameter format?
A: Need to match with regular expression ^[a-zA-Z0-9-_]{32,36}$, and it’s recommended to use uuid to ensure uniqueness.

Q) Place order return error response "Unable to place order due to incorrect price or amount".
A: Please check the order price. You can query the last trade price with Symbol Price Ticker API. Then you can place an order with a more appropriate price. The price range is determined by symbol multiplierUp and multiplierDown from the Exchange Info API
Upper price bound will be lastTradePrice * multiplierUp
Lower price bound will be lastTradePrice * multiplierDown
Generally multiplierUp set to 1000 and multiplierDown set to 0.001
Please also check the quantity can be divided by stepSize from the Exchange Info API

Q) Place order return error response "Invalid token."
A: Please check the request whether baseAsset and quoteAsset are correct, and make sure baseAsset can be found in the get exchange info API. For example
POST
/sapi/v1/alpha-trade/order/place?clientOrderId=4c0e5cde2e9f4b5d8a7b0fbe9f39e5a3&price=0.00100000&quantity=0.10&side=BUY&timestamp=1746692168525&baseAsset=ALPHA_105USDT&quoteAsset=USDT&signature=04cd581219454f20430f81ef3946a5fcc8932cb095f1159728934ce2466e4c57
The above request is an incorrect request, the correct one should be baseAsset=ALPHA_105.

Q) Where can I find all the alpha asset id to actual token mapping?
A: https://www.binance.com/bapi/defi/v1/public/wallet-direct/buw/wallet/cex/alpha/all/token/list

Q) Will alpha asset id remap to another token in the future? For example, today ALPHA_180 represents PUFFER but in the future, it may represent some different token XYZ.
A: No.